UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/740,601	12/19/2000	Joe D. Bolding	10003151-1	3782

7590 04/08/2005

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, CO 80527-2400

EXAMINER	
ROCHE, TRENTON J	

ART UNIT	PAPER NUMBER
2193	

DATE MAILED: 04/08/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/740,601	BOLDING ET AL.	
	Examiner	Art Unit	
	Trent J Roche	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 December 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-25* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-25* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on *19 December 2000* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This office action is responsive to communications filed 10 December 2004.

2. Per applicant's request, amended claims 1, 16, 23 and 25 have been entered. Claims 1-25 are now pending.

3. Claims 1-25 have been examined.

Claim Rejections - 35 USC § 101

4. In view of the applicant's amendments, the rejection of claims 1-25 under 35 U.S.C. § 101 has been withdrawn.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1-6, 13-19, 22 and 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Japanese Patent Publication 63-223927 by Hashimoto in view of "How Debuggers Work" by Rosenberg.

***Note*: all page numbers indicated in Hashimoto are in reference to the English translation document, provided by The Ralph Mcelroy Translation Company**

Regarding claim 1:

Hashimoto discloses:

- A method of debugging ("performs debugging..." on page 1)
- providing a plurality of symbol tables ("the plural symbol tables..." on page 3. Further, note Figure 1, wherein a plurality of symbol tables 101, 102 and 103 are shown.)
- in a computer system ("a system that performs debugging of plural programs" on page 1)
- having an address pointer ("program identification information that uniquely identifies plural programs" on page 2.)
- said symbol tables encompassing a range of addresses (Note Figure 1. The symbol tables include a range of symbols.)
- each of said symbol tables being associated with at least one symbol in common but with different memory offsets ("the various fields of symbol table 101...are taken as already having values contained in them. Also, symbol tables (102), (103) are also taken as containing the same values." on page 3. Further, they must inherently have different memory offsets, or the symbol tables would overlap and there would no longer be multiple symbol tables.)
- selecting at least one of the plurality of symbol tables within whose range of addresses the address pointer is pointing ("the program identification information field in the plural symbol tables is retrieved, and the program identification information field in agreement with program identifier 109 is determined." on page 3)
- wherein said at least one symbol table is selected based on said address pointer ("Then, the symbol field in the symbol table containing the determined program identification information field is retrieved..." on page 3. Further, even when the symbol field is retrieved using the program identification information field, which is a program name, an address

pointer is still inherently used, as an instruction must be issued to retrieve the symbol field which corresponds to the program identification information field. This instruction would contain an address pointer.)

substantially as claimed. While Hashimoto discloses the use of a debugger, it is not explicitly stated that upon accessing the symbol table, displaying at least one symbol while debugging. However, Rosenberg discloses that debuggers in general utilize symbol tables to show and report information to the user concerning current source statements, variables and procedures. ("When the user asks to evaluate an expression, the debugger must use the appropriate scope...the debugger must show the scope or procedure corresponding to each return address...When the program stops, the debugger must show the current source statement...When the debugger disassembles instructions, it may wish to show the names of the variables and procedures to which they refer..." on page 161.) It would have been obvious to one of ordinary skill in the art at the time the invention was made to display at least one symbol while debugging software associated with a plurality of symbol tables in the multiple symbol table system of Hashimoto, as this enable a software developer to "track down, isolate, and remove bugs from software programs..." by allowing "the programmer to follow the flow of program execution and, at any desired point, stop the program and inspect the state of the program to verify its correctness." (note pages 1 and 2 of Rosenberg)

Regarding claim 2:

The rejection of claim 1 is incorporated, and further, Hashimoto discloses a debugger performing said selecting of symbol tables as claimed ("in a system that performs debugging of plural programs...it is possible to make reference to the symbol table with respect to the assigned program" on page 2)

Regarding claim 3:

The rejection of claim 2 is incorporated, and further, Hashimoto discloses the selecting being performed each time a debugger transitions from an executing mode to a command mode as claimed ("By means of program identifier 109 assigned at the same time with symbol 110 in the command with respect to debug, the program identification information field in the plural symbol tables is retrieved, and the program identification information field in agreement with program identifier 109 is determined." on page 3)

Regarding claim 4:

The rejection of claim 1 is incorporated, and further, Hashimoto discloses a computer system performing the selecting of at least one of said plurality of symbol tables as claimed ("in a system that performs debugging of plural programs...it is possible to make reference to the symbol table with respect to the assigned program" on page 2)

Regarding claim 5:

The rejection of claim 1 is incorporated, and further, Hashimoto discloses a pointer to a memory location containing instructions to be executed as claimed ("Program identifier 109 holds the value in agreement with the value of the program identification information field of certain symbol table" on page 3)

Regarding claim 6:

The rejection of claim 5 is incorporated, and further, Hashimoto discloses a program counter as claimed ("in a system that performs debugging of plural programs...it is possible to make reference to the symbol table with respect to the assigned program" on page 2. Since the system is running a plurality of programs, a program counter must inherently be used for the identification of the pertinent symbol table.)

Regarding claim 13:

The rejection of claim 1 is incorporated, and further, Hashimoto discloses selecting a symbol table by marking as active as claimed ("to make reference to the symbol table with respect to the assigned program" on page 1. The system would inherently mark the associated symbol table when it determines which table correctly corresponds to the program.)

Regarding claim 14:

The rejection of claim 13 is incorporated, and further, Hashimoto discloses a debugger using the symbol table as claimed (note the rejection regarding claim 2)

Regarding claim 15:

The rejection of claim 1 is incorporated, and further, Hashimoto discloses the computer system comprising an architectural simulator as claimed ("in a system that performs debugging of plural programs...it is possible to make reference to the symbol table with respect to the assigned program" on page 2. The debugger is simulating the execution of the program.)

Regarding claim 16:

Hashimoto discloses:

- an apparatus for debugging software ("a system that performs debugging...
- at least one computer readable storage medium ("plural programs in the memory" on page 2)
- computer readable program code stored on the at least one computer readable storage medium ("plural programs in the memory" on page 2)
- code for selecting one of a plurality of symbol tables if a program counter in a computer contains an address within said one of said plurality of symbol tables ("in a system that performs debugging of plural programs...it is possible to make reference to the symbol table with respect to the assigned program" on page 2. Further, "Then, the symbol field in the symbol table containing the determined program identification information field is retrieved..." on page 3. Even when the symbol field is retrieved using the program identification information field, which is a program name, an address pointer is still inherently used, as an instruction must be issued to retrieve the symbol field which corresponds to the program identification information field. This instruction would contain an address pointer, and the instruction would come from the program counter, which keeps track of the currently executing instruction in the system.)

substantially as claimed. While Hashimoto discloses the use of a debugger, it is not explicitly stated that upon accessing the symbol table, displaying at least one symbol while debugging. However, Rosenberg discloses that debuggers in general utilize symbol tables to show and report information to the user concerning current source statements, variables and procedures. ("When the user asks to evaluate an expression, the debugger must use the appropriate scope...the debugger must show the scope or procedure corresponding to each return address...When the program stops, the debugger

must show the current source statement...When the debugger disassembles instructions, it may wish to show the names of the variables and procedures to which they refer..." on page 161.) It would have been obvious to one of ordinary skill in the art at the time the invention was made to display at least one symbol while debugging software associated with a plurality of symbol tables in the multiple symbol table system of Hashimoto, as this enable a software developer to "track down, isolate, and remove bugs from software programs..." by allowing "the programmer to follow the flow of program execution and, at any desired point, stop the program and inspect the state of the program to verify its correctness." (note pages 1 and 2 of Rosenberg)

Regarding claim 17:

The rejection of claim 16 is incorporated, and further, Hashimoto discloses determining whether said program counter contains an address within said address range for said one of said plurality of symbol tables as claimed ("the program identification information field in the plural symbol tables is retrieved, and the program identification information field in agreement with program identifier 109 is determined." on page 3. Further, the symbol tables contain ranges of symbols.)

Regarding claim 18:

The rejection of claim 16 is incorporated, and further, Hashimoto discloses determining whether said program counter contains an address within a base symbol table as claimed (Note Figure 1. Further, "the program identification information field in the plural symbol tables is retrieved, and the program identification information field in agreement with program identifier 109 is determined." on page 3. The system checks symbol table 101 before checking symbol tables 102 and 103.)

Regarding claim 19:

The rejection of claim 16 is incorporated, and further, Hashimoto discloses determining whether said program counter contains an address within an offset symbol table as claimed (Note Figure 1. Further, "the program identification information field in the plural symbol tables is retrieved, and the program identification information field in agreement with program identifier 109 is determined." on page 3. The system checks symbol table 101 before checking offset symbol tables 102 and 103.)

Regarding claim 22:

The rejection of claim 16 is incorporated, and further, Hashimoto discloses determining whether said one of said plurality of symbol tables is enabled for automatic selection as claimed ("at start of debugging of the program, by reading the symbol table into the debugger by means of the debugger command..." on page 3. The debugger would inherently make the determination.)

Regarding claim 23:

Hashimoto discloses:

- a debugging apparatus, comprising a computer having a plurality of symbol tables stored thereon, a debugger connected to said computer ("in a system that performs debugging of plural programs...it is possible to make reference to the symbol table with respect to the assigned program" on page 2. Further, "the plural symbol tables..." on page 3. Note Figure 1, wherein a plurality of symbol tables 101, 102 and 103 are shown.)

- means for automatically selecting at least one of said plurality of symbol tables in said computer for said debugger ("in a system that performs debugging of plural programs...it is possible to make reference to the symbol table with respect to the assigned program" on page 2. Further, "Then, the symbol field in the symbol table containing the determined program identification information field is retrieved..." on page 3. Even when the symbol field is retrieved using the program identification information field, which is a program name, an address pointer is still inherently used, as an instruction must be issued to retrieve the symbol field which corresponds to the program identification information field. This instruction would contain an address pointer.)

substantially as claimed. While Hashimoto discloses the use of a debugger, it is not explicitly stated that upon accessing the symbol table, displaying at least one symbol while debugging. However, Rosenberg discloses that debuggers in general utilize symbol tables to show and report information to the user concerning current source statements, variables and procedures. ("When the user asks to evaluate an expression, the debugger must use the appropriate scope...the debugger must show the scope or procedure corresponding to each return address...When the program stops, the debugger must show the current source statement...When the debugger disassembles instructions, it may wish to show the names of the variables and procedures to which they refer..." on page 161.) It would have been obvious to one of ordinary skill in the art at the time the invention was made to display at least one symbol while debugging software associated with a plurality of symbol tables in the multiple symbol table system of Hashimoto, as this enable a software developer to "track down, isolate, and remove bugs from software programs..." by allowing "the programmer to follow the flow of program execution and, at any desired point, stop the program and inspect the state of the program to verify its correctness." (note pages 1 and 2 of Rosenberg)

7. Claims 7-12, 20, 21, 24 and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Japanese Patent Publication 63-223927 by Hashimoto in view of "How Debuggers Work" by Rosenberg, further in view of U.S. Patent 5,805,867 to Kodaira.

Per claim 7:

The rejection of claim 1 is incorporated, and further, Hashimoto discloses a symbol table being selected for an active program. Neither Hashimoto nor Rosenberg explicitly disclose a plurality of cells, each of said cell comprising a processing unit having at least one computer processor. Kodaira discloses in an analogous debugging and simulation system the use of multiple processors in a system as claimed ("The processing speed of the entire multi-processor system can be increased when parallel processing is conducted at the same time by a plurality of processors" in col. 1 lines 19-21). It would have been obvious to one of ordinary skill in the art at the time the invention was made to use multiple processors in the debugging system disclosed by Hashimoto modified by Rosenberg, as this would enable the user to experience an increase in processing speed in the debugging system disclosed by Hashimoto, as disclosed by Kodaira in col. 1 lines 19-21.

Per claim 8:

The rejection of claim 7 is incorporated, and further, Hashimoto discloses examining said at least one base symbol table to determine whether said address pointer is pointing within said at least one base symbol table, and examining at least one of said plurality of secondary symbol tables to determine whether said address pointer is pointing within said at least one of said plurality of secondary symbol tables as claimed (Note Figure 1. Further, "the program identification information

field in the plural symbol tables is retrieved, and the program identification information field in agreement with program identifier 109 is determined." on page 3. The system checks symbol table 101 before checking offset symbol tables 102 and 103.)

Per claim 9:

The rejection of claim 8 is incorporated, and further, Hashimoto discloses wherein each of said plurality of secondary symbol tables comprises a reference to a base symbol table, a cell identifier, and an address offset specifying an offset from said base symbol table as claimed (Note Figure 1 and the corresponding sections of the disclosure)

Per claim 10:

The rejection of claim 8 is incorporated, and further, Hashimoto discloses wherein at least one base symbol table is examined before said at least one of said plurality of secondary symbol tables is examined as claimed (Note Figure 1. Further, "the program identification information field in the plural symbol tables is retrieved, and the program identification information field in agreement with program identifier 109 is determined." on page 3. The system checks symbol table 101 before checking offset symbol tables 102 and 103.)

Per claim 11:

The rejection of claim 8 is incorporated, and further, Hashimoto discloses wherein at least one of said plurality of secondary symbol tables is only examined if said address pointer is not pointing within said at least one base symbol table as claimed (Note Figure 1. Further, "the program identification information field in the plural symbol tables is retrieved, and the program

identification information field in agreement with program identifier 109 is determined." on page 3. The system checks symbol table 101 before checking offset symbol tables 102 and 103.)

Per claim 12:

The rejection of claim 8 is incorporated, and further, Hashimoto discloses examining at least one of said plurality of secondary symbol tables comprises checking a cell identifier within each of said plurality of secondary symbol tables to determine whether each of said plurality of second symbol tables is associated with that active cell, and examining only tables within said plurality of second symbol tables which are associated with said active cell to determine whether said tables which are associated with said active cell should be selected as claimed ("By means of program identifier 109 assigned at the same time with symbol 110 in the command with respect to debug, the program identification information field in the plural symbol tables is retrieved, and the program identification information field in agreement with program identifier 109 is determined." on page 3)

Per claim 20:

The rejection of claim 19 is incorporated, and further, note the rejection regarding claim 7.

Per claim 21:

The rejection of claim 20 is incorporated, and further, note the rejection regarding claim 12.

Per claim 24:

The rejection of claim 23 is incorporated, and further, note the rejection regarding claim 7.

Per claim 25:

Hashimoto discloses:

- an apparatus for debugging software
- at least one computer readable storage medium ("plural programs in the memory" on page 2)
- computer readable program code stored on the at least one computer readable storage medium ("plural programs in the memory" on page 2)
- code for selecting one of the plurality of symbol tables and using the symbol table with a processing cell ("in a system that performs debugging of plural programs...it is possible to make reference to the symbol table with respect to the assigned program" on page 2)
- wherein said at least one symbol table is selected if a program counter in said computer contains an address within said at least one symbol table ("in a system that performs debugging of plural programs...it is possible to make reference to the symbol table with respect to the assigned program" on page 2. Further, "Then, the symbol field in the symbol table containing the determined program identification information field is retrieved..." on page 3. Even when the symbol field is retrieved using the program identification information field, which is a program name, an address pointer is still inherently used, as an instruction must be issued to retrieve the symbol field which corresponds to the program identification information field. This instruction would contain an address pointer, and the instruction would come from the program counter, which keeps track of the currently executing instruction in the system.)

substantially as claimed. While Hashimoto discloses the use of a debugger, it is not explicitly stated that upon accessing the symbol table, displaying at least one symbol while debugging. However,

Rosenberg discloses that debuggers in general utilize symbol tables to show and report information to the user concerning current source statements, variables and procedures. ("When the user asks to evaluate an expression, the debugger must use the appropriate scope...the debugger must show the scope or procedure corresponding to each return address...When the program stops, the debugger must show the current source statement...When the debugger disassembles instructions, it may wish to show the names of the variables and procedures to which they refer..." on page 161.) It would have been obvious to one of ordinary skill in the art at the time the invention was made to display at least one symbol while debugging software associated with a plurality of symbol tables in the multiple symbol table system of Hashimoto, as this enable a software developer to "track down, isolate, and remove bugs from software programs..." by allowing "the programmer to follow the flow of program execution and, at any desired point, stop the program and inspect the state of the program to verify its correctness." (note pages 1 and 2 of Rosenberg)

Further, neither Hashimoto nor Rosenberg explicitly disclose a plurality of processing cells. Kodaira discloses in an analogous debugging and simulation system the use of multiple processors in a system as claimed ("The processing speed of the entire multi-processor system can be increased when parallel processing is conducted at the same time by a plurality of processors" in col. 1 lines 19-21). It would have been obvious to one of ordinary skill in the art at the time the invention was made to use multiple processors in the debugging system disclosed by Hashimoto modified by Rosenberg, as this would enable the user to experience an increase in processing speed in the debugging system disclosed by Hashimoto, as disclosed by Kodaira in col. 1 lines 19-21.

Response to Arguments

8. Applicant's arguments filed 10 December 2004 have been fully considered but they are not persuasive.

Per claims 1, 16, 23 and 25:

The applicant states that Hashimoto does not disclose or suggest the use of multiple symbol tables, all associated with at least one symbol in common but with different memory offsets, and selecting one of the symbol tables based on an address pointer. In response, the Examiner contends that the multiple symbol tables of Hashimoto are indeed associated with at least one symbol in common, as noted on page 3 wherein "symbol tables (102), (103) are also taken as containing the same values [of symbol table 101]..." Furthermore, the tables must be at different memory offsets, otherwise the multiple symbol tables would not exist and would overlap each other. Furthermore, in regards to selecting one of the symbol tables based on an address pointer, Hashimoto discloses that "the symbol field in the symbol table containing the determined program identification information field is retrieved, and the symbol field in agreement with symbol 110 in the command is selected..." (page 3) The Examiner contends that, while the program identification information field may be a program name, for the computer system to physically retrieve the symbol from the symbol table associated with the identification information field, an instruction must be issued to select and retrieve the symbol. As a program counter of a system is a register that contains the address of the instruction to be executed next, then the address pointer for retrieving the symbol would come from the program counter if the retrieval instruction were to execute. As such, Hashimoto discloses the required limitations, and the rejections of claim 1, 16, 23 and 25 are proper and maintained.

Per claims 2-15, 17-22 and 24:

The applicant states that claims 2-15, 17-22 and 24 are allowable as being dependent on an allowable base claim. As was shown above, Hashimoto discloses all the required limitations of the independent claims, and as such, the argument that claims 2-15, 17-22 and 24 are allowable as being dependent on an allowable base claim is considered moot.

Conclusion

9. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Trent J Roche whose telephone number is (571) 272-3733. The examiner can normally be reached on Monday - Friday, 9:00 am - 5:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571) 272-3719. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Trent J Roche
Examiner
Art Unit 2193

TJR

Kakali Chaki

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100